On the circuit-size of inverses

J.C. Birget

February 25, 2011

Abstract

We reprove a result of Boppana and Lagarias: If $\Pi_2^{\mathsf{P}} \neq \Sigma_2^{\mathsf{P}}$ then there exists a partial function f that is computable by a polynomial-size family of circuits, but no inverse of f is computable by a polynomial-size family of circuits. We strengthen this result by showing that there exist length-preserving total functions that are one-way by circuit size and that are computable in *uniform* polynomial time. We also prove, if $\Pi_2^{\mathsf{P}} \neq \Sigma_2^{\mathsf{P}}$, that there exist polynomially balanced total *surjective* functions that are one-way by circuit size; here non-uniformity is used.

KEYWORDS: Computational complexity, circuit size, one-way functions

1 Introduction

The difficulty of inversion (i.e., given f and y, find any x such that $f(x) = y$) is a fundamental topic in computational complexity and in cryptography. The question whether NP is different from P can be formulated as a question about the difficulty of inversion, namely, $\mathsf{P} \neq \mathsf{NP}$ iff there exists a one-way function based on polynomial-time ([11], [8] pp. 32-43, [5] pp. 119-125). A function f is said to be one-way based on polynomial time iff f is polynomial-time computable (by a deterministic Turing machine) but no inverse function f' of f is polynomial-time computable. An *inverse* of f is any function f' that $f \circ f' \circ f = f$. In this paper we consider one-way functions based on (non-uniform) families of circuits of polynomial size. Boppana and Lagarias [2] (by using the Karp-Lipton theorem [9]) proved that if $\Pi_2^{\mathsf{P}} \neq \Sigma_2^{\mathsf{P}}$ then there exists a partial function f that can be computed by a non-uniform family of circuits of polynomial size, but no inverse f' of f can be computed by a non-uniform family of circuits of polynomial size. We show that this result still holds when f is a total surjective and polynomially balanced function, or when f is length-preserving and uniformly computable in polynomial time (but non-uniformity is allowed for the inverses).

By "circuit" we mean a digital circuit made of boolean gates, whose underlying directed graph is acyclic [16]. More precisely, a circuit C with m input vertices and n output vertices, consists of two parts. First, C has an acyclic directed graph (with vertex set V and edge set E); we assume that the set of vertices V has a total order (i.e., V is not just a set but a sequence). Second, C has a *gate map*

$$\mathsf{gate}: \quad v \in V \quad \longmapsto \quad \mathsf{gate}(v) \in \{\mathsf{and}, \mathsf{or}, \mathsf{not}, \mathsf{fork}, \mathsf{in}_1, \ldots, \mathsf{in}_m, \mathsf{out}_1, \ldots, \mathsf{out}_n\}$$

which assigns a gate $\mathsf{gate}(v)$ to each vertex v. The gates and, or, and not are the traditional boolean operations. The gates and and or have domain $\{0,1\} \times \{0,1\}$, so a vertex labeled by such a gate has in-degree 2; not has domain $\{0,1\}$, so a vertex labeled by not has in-degree 1; all three operations have codomain $\{0,1\}$, so the vertex has out-degree 1. The gate $\mathsf{fork}: x \in \{0,1\} \mapsto (x,x) \in \{0,1\} \times \{0,1\}$ is also called the fan-out operation; the corresponding vertex has in-degree 1 and out-degree 2. Input vertices are mapped to $\mathsf{in}_1, \ldots, \mathsf{in}_m$; they have in-degree 0 and out-degree 1. Output vertices are mapped to $\mathsf{out}_1, \ldots, \mathsf{out}_n$; they have in-degree 1 and out-degree 0. The gate map is injective on the union of the set of input vertices and the set of output vertices.

The *size* (or complexity) of a circuit C, denoted $|C|$, is defined to be the number edges (i.e., wire links) plus the number of vertices. Thus $|C|$ is always at least as large as the number of input vertices, plus the number of output vertices. A circuit C with m input vertices and n output vertices has an *input-output function*

$(x_1, \ldots, x_m) \in \{0,1\}^m \longmapsto (y_1, \ldots, y_n) \in \{0,1\}^n$ that we denote by $C(.)$. The image set of C, i.e. the set all actual outputs, is denoted by $\mathsf{im}(C)$ ($\subseteq \{0,1\}^n$).

Let A be a finite alphabet; when we talk about circuits we always assume that $A = \{0,1\}$.

Definition 1.1 *A function $f : A^* \to A^*$ is called* length-equality preserving *iff for all $x_1, x_2 \in A^*$, $|x_1| = |x_2|$ implies $|f(x_1)| = |f(x_2)|$. Equivalently, for every m there exists n such that $f(A^m) \subseteq A^n$.*

A special case consists of the length-preserving *functions, satisfying $|f(x)| = |x|$.*

Definition 1.2 *A function $f : A^* \to A^*$ is called* polynomially balanced *iff there exist polynomials $p_1(.)$ and $p_2(.)$ such that for all inputs $x \in A^*$: $|f(x)| \leq p_1(|x|)$ and $|x| \leq p_2(|f(x)|)$.*

A special case is, again, the length-preserving functions.

Definition 1.3 *A length-equality preserving function $f : \{0,1\}^* \to \{0,1\}^*$ is said to be* computed by a family of circuits $\mathbf{C} = \{C_m : m \in \mathbb{N}\}$ *iff for all $m \in \mathbb{N}$ and all $x \in \{0,1\}^m$, $f(x) = C(x)$. (We do not make any uniformity assumptions for* **C**.*)*

This family is said to be of polynomial size *iff there is a polynomial $p(.)$ such that for all m : $|C_m| \leq p(m)$.*

In general, a family of circuits $\mathbf{C} = \{C_i : i \in \mathbb{N}\}$ could contain any number of circuits C_i with the same number of input vertices; then **C** does not compute a function.

Computational *one-wayness* can be defined in many (non-equivalent) ways. We will use the following definition, related to worst-case circuit complexity (we are not considering cryptographic one-way functions here).

Definition 1.4 *A length-equality preserving function $f : \{0,1\}^* \to \{0,1\}^*$ is* one-way by circuit size *iff*

- *f is polynomially balanced,*
- *f is computable by a polynomial-size family of circuits, but*
- *no inverse function f' of f is computable by a polynomial-size family of circuits.*

Intuitively, one-wayness based on circuit size should be stronger than one-wayness based on uniform computational complexity. Indeed, in the former, not only is it difficult to find any inverse f' of f, but the circuits for the inverses f' are all very large. Definition 1.4 can also be adapted to a family of circuits, by itself.

Definition 1.5 *A family of circuits $\mathbf{C} = \{C_i : i \in \mathbb{N}\}$ is* one-way by circuit size *iff for every polynomial $p(.)$ there is* no *family of circuits $\mathbf{C}' = \{C'_i : i \in \mathbb{N}\}$ such that for all i, $C_i \circ C'_i \circ C_i(.) = C_i(.)$ and $|C'_i| \leq p(|C_i|)$.*

Before dealing with one-wayness we characterize the complexity of the injectiveness problem and of the surjectiveness problem for circuits. Injectiveness is equivalent to the existence of left inverses, and surjectiveness is equivalent to the existence of right inverses. After that we consider general inverses.

2 Injectiveness and surjectiveness

The *equivalence problem* for circuits takes two circuits C_1, C_2 as input, and asks whether $C_1(.) = C_2(.)$. It is well known that this problem is coNP-complete [5, 8]. A related problem is the following, where for any set S we denote the identity function on S by id_S. In the *identity problem*, for a given circuit C the question is whether $C(.) = \mathsf{id}_{\{0,1\}^n}$. In the *injectiveness problem* the question is whether $C(.)$ is injective. The identity problem is a special case of both the equivalence problem and the injectiveness problem.

Proposition 2.1 *The injectiveness problem and the identity problem for circuits are* coNP*-complete.*

Proof (this is Theorem 6.5 in [1], reproved here purely in the context of circuits). It is easy to see that the injectiveness problem and the identity problem are in coNP. To show hardness we reduce the tautology problem for boolean formulas to the injectiveness problem and to the identity problem for circuits, as follows. Let B be any boolean formula with n variables. We define a new boolean function $F_B : \{0,1\}^{n+1} \to \{0,1\}^{n+1}$ by

$$F_B(x_1, \ldots, x_n, x_{n+1}) = \begin{cases} (x_1, \ldots, x_n, x_{n+1}) & \text{if } B(x_1, \ldots, x_n) = 1 \text{ or } x_{n+1} = 1, \\ (1, \ldots, 1, 1) \ (= 1^{n+1}) & \text{otherwise.} \end{cases}$$

Let us check that the following three properties are equivalent: (1) B is a tautology, (2) F_B is injective, and (3) $F_B = \mathsf{id}_{\{0,1\}^{n+1}}$.

When $B(x_1, \ldots, x_n) = 1$ then $F_B(x_1, \ldots, x_n, x_{n+1}) = (x_1, \ldots, x_n, x_{n+1})$. So, if B is a tautology then F_B is the identity function on $\{0,1\}^{n+1}$ (which also implies that F_B is injective).

If B is a not a tautology then $B(c_1, \ldots, c_n) = 0$ for some $(c_1, \ldots, c_n) \in \{0,1\}^n$. It follows that $F_B(c_1, \ldots, c_n, 0) = (1, \ldots, 1, 1)$. But we also have $F_B(1, \ldots, 1, 1) = (1, \ldots, 1, 1)$, since here $x_{n+1} = 1$. Hence, F_B is not injective (and hence not the identity function). □.

The *surjectiveness problem* for circuits takes a circuit C as input, and asks whether $C(.)$ is surjective. Let Π_2^{P} denote the $\forall\exists$-class at level 2 in the polynomial hierarchy [5, 8]; similarly, Σ_2^{P} denotes the $\exists\forall$-class. Theorem 2.2 below is very similar to Theorem 5.9 in [1] about the surjectiveness problem for elements of the Thompson-Higman monoid $M_{2,1}$. But there are technical differences between circuits and elements of $M_{2,1}$, so we give a separate proof for circuits here.

Theorem 2.2 *The surjectiveness problem for circuits is* Π_2^{P}*-complete.*

Proof. The definition of surjectiveness shows that the surjectiveness problem is in Π_2^{P}. Indeed, $C(.)$ is surjective iff $(\forall y \in \{0,1\}^n)(\exists x \in \{0,1\}^m)\,[C(x) = y]$. This is a Π_2^{P}-formula, since $n, m \leq |C|$, and since the property $C(x) = y$ can be checked deterministically in polynomial time when x, y, and C are given.

Let us prove hardness by reducing $\forall\exists$Sat (the $\forall\exists$-satisfiability problem) to the surjectiveness problem for circuits. Let $B(x, y)$ be any boolean formula where x is a sequence of m boolean variables, and y is a sequence of n boolean variables. The problem $\forall\exists$Sat asks on input $\forall y \exists x\, B(x, y)$ whether this sentence is true. It is well known that $\forall\exists$Sat is Π_2^{P}-complete [5, 8]. We map the formula B to the circuit C_B with input-output function defined by

$$C_B(x, y, y_{n+1}) = \begin{cases} (y, y_{n+1}) & \text{if } B(x,y) = 1 \text{ or } y_{n+1} = 1, \\ (1^n, 1) & \text{if } B(x,y) = y_{n+1} = 0. \end{cases}$$

Equivalently,

$$C_B(x, y, y_{n+1}) = (y_1 \vee \overline{(B(x,y) \vee y_{n+1})},\ \ldots,\ y_n \vee \overline{(B(x,y) \vee y_{n+1})},\ y_{n+1} \vee \overline{B(x,y)}\,) .$$

Hence one can easily construct a circuit for C_B from the formula $B(x, y)$. By the definition of C_B ,

$$\mathsf{im}(C_B) = \{(y, 0) : \exists x B(x, y)\} \cup \{(y, 1) : y \in \{0,1\}^n\} \quad (\cup \{(1^n, 1)\}) .$$

Since $(1^n, 1) \in \{(y, 1) : y \in \{0,1\}^n\}$, the term $\{(1^n, 1)\}$ (which may or may not be present) is irrelevant. Hence,

$$\mathsf{im}(C_B) = \{0,1\}^n 1 \cup \{y \in \{0,1\}^n : \exists x B(x, y)\}\, 0 .$$

Therefore, $\forall y \exists x\, B(x, y)$ is true iff $\mathsf{im}(C_B) = \{0,1\}^n 1 \cup \{0,1\}^n 0$, i.e., iff C_B is surjective. □

For a partial function $f : X \to Y$ it is a well-known fact that f is surjective iff f has a right inverse. By definition, a partial function $g : Y \to X$ is called a *right inverse* of f iff $f \circ g(.) = \mathsf{id}_Y$. For circuits we have: A circuit C (with m input wires and n output wires) is surjective iff there exists a circuit C' (with n input wires and m output wires) such that $C \circ C'(.) = \mathsf{id}_{\{0,1\}^n}$.

Theorem 2.3 *If there exists a polynomial* $p(.)$ *such that every surjective circuit* C *has a* right inverse C' *of size* $|C'| \leq p(|C|)$*, then* $\Pi_2^{\mathsf{P}} = \Sigma_2^{\mathsf{P}}$.

Proof. If such a polynomial $p(.)$ exists then the surjectiveness of C is characterized by

$$C \text{ is surjective} \quad \text{iff} \quad (\exists C',\ |C'| \leq p(|C|))\,(\forall x \in \{0,1\}^m)\ [C \circ C'(x) = x].$$

This is a Σ_2^{P}-formula since the quantified variables are polynomially bounded in terms of $|C|$, and the relation $C \circ C'(x) = x$ can be checked deterministically in polynomial time when C, C' and x are given. This implies that the surjectiveness problem is in Σ_2^{P}. But since we already proved that the surjectiveness problem is Π_2^{P}-complete, this implies that $\Pi_2^{\mathsf{P}} \subseteq \Sigma_2^{\mathsf{P}}$. Hence, $\Pi_2^{\mathsf{P}} = \Sigma_2^{\mathsf{P}}$. □

3 General inverses

The general concept of an inverse goes back to Moore [12] (Moore-Penrose pseudo-inverse of a matrix), and von Neumann [13] (regular rings). For a partial function $f : X \to Y$, the domain of f is denoted by $\mathsf{dom}(f)$ ($\subseteq X$), and the image (or range) is denoted by $\mathsf{im}(f)$ ($\subseteq Y$). A partial function $f : X \to Y$ is called *total* iff $\mathsf{dom}(f) = X$. When we just say "function" we mean a total function.

Definition 3.1 *For a partial function $F : X \to Y$ an* inverse *(also called a* semi-inverse*) of F is any partial function $F' : Y \to X$ such that $F \circ F' \circ F = F$. If both $F \circ F' \circ F = F$ and $F' \circ F \circ F' = F'$ hold then F' is a* mutual inverse *of F, and F is a mutual inverse of F'.*

The following facts about inverses are well known and straightforward to prove. For any two partial functions $F : X \to Y$ and $F' : Y \to X$ we have:

- *$F \circ F' \circ F = F$ iff $(F \circ F')_{\mathsf{im}(F)} = \mathsf{id}_{\mathsf{im}(F)}$, where $(.)_{\mathsf{im}(F)}$ denotes the restriction to $\mathsf{im}(F)$.*
- *If F' is a semi-inverse of F then $\mathsf{im}(F) \subseteq \mathsf{dom}(F')$; i.e., $F'(y)$ is defined for all $y \in \mathsf{im}(F)$.*
- *If F' is a semi-inverse of F then $F'_{\mathsf{im}(F)}$ is injective.*
- *If F' is a semi-inverse of F then $F' \circ F \circ F'$ is a mutual inverse of F.*
- *Every partial function F has at least one semi-inverse. More specifically, F has at least one semi-inverse F'_1 that is total (i.e., $\mathsf{dom}(F'_1) = Y$), and at least one semi-inverse F'_2 that is injective and whose domain is $\mathsf{im}(F)$.*

For infinite sets the last fact requires the axiom of choice. The following two Lemmas are also straightforward.

Lemma 3.2 *F' is a right inverse of F iff F' is a total and injective mutual inverse of F.* □

Lemma 3.3 *For a partial function $F : X \to Y$ the following are equivalent:*

(1) *F is* surjective*;*

(2) *F has a right inverse;*

(3) *F has a mutual inverse F' that is total and injective;*

(4) *every semi-inverse F' of F is total and injective;*

(5) *every semi-inverse F' of F is total.* □

We can now reformulate Theorem 2.3 in terms of inverses.

Theorem 3.4 *If there exists a polynomial $p(.)$ such that every circuit C has a semi-inverse C' of size $|C'| \leq p(|C|)$, then $\Pi_2^{\mathsf{P}} = \Sigma_2^{\mathsf{P}}$.*

Proof. If such a $p(.)$ exists then every circuit C has an inverse C' of size $|C'| \leq p(|C|)$, and hence every C has a mutual inverse $C'_2 = C' \circ C \circ C'$ of size $|C'_2| \leq 2 \cdot p(|C|) + |C|$. Let $q(n) = 2 \cdot p(n) + n$, which is also a polynomial.

Let us now consider the special case where C is surjective. Then by Lemma 3.3(1 $\Rightarrow$ 4), C'_2 is total and injective. Then by Lemma 3.2, since C'_2 is a mutual inverse, C'_2 is a right inverse of C. Now Theorem 2.3 (for the polynomial $q(.)$) implies that $\Pi_2^{\mathsf{P}} = \Sigma_2^{\mathsf{P}}$. □

Theorem 3.4 is not new; it follows immediately from a result by Boppana and Lagarias (Theorem 2.1a in [2]), combined with the Karp-Lipton Theorem [9, 5, 8].

The proof of Theorem 3.4 also applies to surjective functions (while the methods in [2] do not seem to):

Corollary 3.5 *If there exists a polynomial $p(.)$ such that every* surjective *circuit C has a* semi-inverse *C' of size $|C'| \leq p(|C|)$, then $\Pi_2^{\mathsf{P}} = \Sigma_2^{\mathsf{P}}$.* □

Theorems 2.3, 3.4 and Coroll. 3.5 show that the family of all circuits and the family of all surjective circuits are one-way by circuit-size.

4 One-way functions, if $\Pi_2^{\mathsf{P}} \neq \Sigma_2^{\mathsf{P}}$

We will use the above results to construct two types of functions that are one-way by circuit-size.

4.1 A surjective non-uniform one-way function

The papers [6] and [4] discuss the existence of surjective one-way functions, based on uniform polynomial time complexity. In the uniform case (with uniformity for both f and f'), it is known that $\mathsf{P} \neq \mathsf{NP} \cap \mathsf{coNP}$ implies the existence of one-way functions (attributed to [3] in the Introduction of [6]). Here we give an existence result for surjective one-way functions with respect to non-uniform polynomial time, i.e., circuit size.

For a circuit C we will denote the number of input vertices by m_C or m, and the number of output wires by n_C or n. An *identity wire* in a circuit is an edge (x_i, y_j) that directly connects an input vertex x_i to an output vertex y_j; so x_i and y_j have the same value. To add an identity wire means to create a new input vertex, a new output vertex, and an edge between them.

Lemma 4.1 *Suppose C_0 is obtained from C by adding identity wires. Then C_0 is surjective iff C is surjective.*

Proof. Let j be the number of identity wires added. So, $\mathsf{im}(C_0) = \mathsf{im}(C) \times \{0,1\}^j$. Then C is surjective iff $\mathsf{im}(C) = \{0,1\}^n$ iff $\mathsf{im}(C_0) = \{0,1\}^n \times \{0,1\}^j = \{0,1\}^{n+j}$ iff C_0 is surjective. □

Proposition 4.2 *Theorem 2.3 and Corollary 3.5 still hold when one only considers surjective circuits C that satisfy $m \leq \frac{1}{2}|C| < 2n$. The same holds if one considers only surjective circuits that satisfy $2n < m \leq |C| < 6n$.*

Proof. From any circuit C one can construct a circuit C_1 by adding $|C|$ identity wires. Then C is surjective iff C_1 is surjective (by Lemma 4.1). An identity wire has two vertices and one edge, so the resulting circuit C_1 has size $|C_1| = 4\,|C|$. For the number of input vertices and output vertices we have $m_1 = m + |C|$, and $n_1 = n + |C|$. Since $m \leq |C|$, it follows that $m_1 \leq \frac{1}{2}|C_1|$. Also, $|C_1| = 4\,(n_1 - n) < 4\,n_1$.

The circuit C_1 satisfies $2n_1 > m_1$ (since $m_1 \leq \frac{1}{2}|C_1| < 2n_1$). Now $2n_1 - m_1 + 1$ new input vertices can be added to C_1; these vertices are not connected to anything and are not output vertices. Then the new circuit C_2 is surjective iff C_1 is surjective. The new circuit C_2 satisfies $n_2 = n_1$, $|C_2| = |C_1| + 2n_1 - m_1 + 1 \leq |C_1| + 2n_1 < 4n_1 + 2n_1$, and $m_2 = 2n_1 + 1 > 2\,n_2$. Hence, $2\,n_2 < m_2 \leq |C_2| \leq 6\,n_2$.

The circuits C_1 and C_2 can be constructed from C deterministically in polynomial time. Moreover, an inverse of C can be obtained in polynomial time from an inverse of C_1, and vice versa. The same holds for C_2. Hence, C has an inverse of size $\leq p(|C|)$ (for some polynomial $p(.)$) iff C_i has an inverse of size $\leq p_i(|C_i|)$ (for some polynomial $p_i(.)$, $i = 1, 2$). Since the existence of polynomial-size inverses for all surjective circuits C implies $\Pi_2^{\mathsf{P}} = \Sigma_2^{\mathsf{P}}$ (by Corollary 3.5), the existence of polynomial-size inverses for C_1 or C_2 also implies $\Pi_2^{\mathsf{P}} = \Sigma_2^{\mathsf{P}}$. □

We saw in Lemma 3.3 that a function $f : X \to Y$ is surjective iff every inverse of f is total and injective.

Theorem 4.3 *For every polynomial $p(.)$ consider the following set of* surjective *circuits:*

$$\mathbf{C}_p \;=\; \{\, C \;:\; 2\,n_C < m_C \le |C| < 6\,n_C \;\text{ and every inverse } C' \text{ of } C \text{ satisfies } |C'| > p(|C|)\,\}.$$

If $\Pi_2^{\mathsf{P}} \neq \Sigma_2^{\mathsf{P}}$ then for every polynomial $p(.)$ the set $\{n_C : C \in \mathbf{C}_p\}$ (consisting of the output lengths of the circuits in $\mathbf{C}_p$) is infinite.

Proof. We assume $\Pi_2^{\mathsf{P}} \neq \Sigma_2^{\mathsf{P}}$. Then by Corollary 3.5 and Prop. 4.2, $\mathbf{C}_p$ is not empty. For all $C \in \mathbf{C}_p$ we have $2\,n_C < m_C \le |C| < 6\,n_C$. It follows that for any polynomial $p(.)$ the four sets $\mathbf{C}_p$, $\{|C| : C \in \mathbf{C}_p\}$, $\{n_C : C \in \mathbf{C}_p\}$, and $\{m_C : C \in \mathbf{C}_p\}$ are all infinite iff one of them is infinite. Moreover, if a function is surjective then all its inverses are total and injective (Lemma 3.3). Hence, $\mathbf{C}_p$ is infinite iff the set $\{C' : C' \textit{ is an inverse of some } C \in \mathbf{C}_p\}$ is infinite, iff the set $\{|C'| : C' \textit{ is an inverse of some } C \in \mathbf{C}_p\}$ is infinite.

For two polynomials we write $p_2 \ge p_1$ when $p_2(n) \ge p_1(n)$ for all n. If $p_2 \ge p_1$ then $\mathbf{C}_{p_2} \subseteq \mathbf{C}_{p_1}$; hence for any polynomial $p_0(.)$ we have $\bigcup_{p \ge p_0} \mathbf{C}_p \;=\; \mathbf{C}_{p_0}$. For any polynomial $p_0(.)$ the set

$$\{p(|C|) :\; p(.) \textit{ is a polynomial, } p \ge p_0, \textit{ and } C \in \mathbf{C}_p\}$$

is infinite; indeed, the set of polynomials is infinite and each $\mathbf{C}_p$ is non-empty. It follows that for any polynomial $p_0(.)$ the set $\{|C'| : C' \textit{ is an inverse of some } C \in \mathbf{C}_p, \textit{ for some } p \ge p_0\}$ is infinite, since $|C'| > p(|C|)$ when $C \in \mathbf{C}_p$. Hence, for any $p_0(.)$, $\mathbf{C}_{p_0}$ and $\{n_C : C \in \mathbf{C}_{p_0}\}$ are infinite. □

Theorem 4.4 *If $\Pi_2^{\mathsf{P}} \neq \Sigma_2^{\mathsf{P}}$ then there exists a* surjective *total function $f : \{0,1\}^* \to \{0,1\}^*$ which is polynomially balanced and length-equality preserving, and which satisfies:*

- *f is computed by a non-uniform polynomial-size family of circuits, but*
- *f has no inverse that can be computed by a non-uniform polynomial-size family of circuits.*

Proof. Consider an infinite sequence of polynomials $p_1 < p_2 < \ldots < p_k < \ldots$, with $p_k(x) > x^k + k$ for all numbers x. Recall that $\mathbf{C}_{p_k} = \{\, C : 2\,n_C < m_C \le |C| < 6\,n_C \textit{ and every inverse } C' \textit{ of } C \textit{ satisfies } |C'| > p_k(|C|)\,\}$. Let us abbreviate $\mathbf{C}_{p_k}$ by $\mathbf{C}_k$. We saw that $\ldots \subseteq \mathbf{C}_k \subseteq \ldots \subseteq \mathbf{C}_2 \subseteq \mathbf{C}_1$. By Theorem 4.3, if $\Pi_2^{\mathsf{P}} \neq \Sigma_2^{\mathsf{P}}$ then $\mathbf{C}_k$ and $\{|C| : C \in \mathbf{C}_k\}$ are infinite for every k. We now construct an infinite set of circuits $\{C_k \in \mathbf{C}_k : k \in \mathbf{N}\}$, where we abbreviate m_{C_k} and n_{C_k} by m_k, respectively n_k.

C_1 is a smallest circuit in $\mathbf{C}_1$;

C_{k+1} is a smallest circuit in $\{C \in \mathbf{C}_{k+1} :\; |C| > |C_k|,\;\; n_C > 1 + n_k \;\text{ and }\; m_C > 2\,m_k\}$.

Since $\mathbf{C}_{k+1}$ is infinite (by Theorem 4.3), the circuit C_{k+1} exists.

Claim: $m_{k+1} - m_k \;>\; n_{k+1} - n_k \;>\; 1$.

Proof of the Claim: We have $m_{k+1} > 2m_k$ (by the choice of C_{k+1}), and $m_{k+1} > 2n_{k+1}$ (since $C_{k+1} \in \mathbf{C}_{k+1}$). Hence, $m_{k+1}/2 > m_k$ and $m_{k+1}/2 > n_{k+1}$. By adding these inequalities we obtain $m_{k+1} > m_k + n_{k+1}$, hence $m_{k+1} - m_k > n_{k+1} > n_{k+1} - n_k$. Also, the choice of $n_C > 1 + n_k$ implies $n_{k+1} - n_k > 1$. This proves the Claim.

We define a total and surjective function $F : \{0,1\}^* \to \{0,1\}^*$ as follows:

(1) $\qquad F(x) \;=\; C_k(x) \;\text{ if }\; |x| = m_k$;

(2) $\qquad F \;\text{ maps }\; D_k = \bigcup_{m=m_k+1}^{m_{k+1}-1} \{0,1\}^m \;\text{ onto }\; R_k = \bigcup_{n=n_k+1}^{n_{k+1}-1} \{0,1\}^n$.

In (1), F maps $\{0,1\}^{m_k}$ onto $\{0,1\}^{n_k}$ for every k, since C_k is surjective. In (2), D_k and R_k are non-empty, since $m_{k+1} - m_k > n_{k+1} - n_k \ge 1$ (by the Claim). To complete the definition of F, D_k can be mapped onto R_k in a length-equality preserving way, as follows: Since $m_{k+1} - m_k - 1 > n_{k+1} - n_k - 1$ and $m_i > n_i$ (for all i), we can map $\{0,1\}^{m_k+i}$ onto $\{0,1\}^{n_k+i}$ for $i = 1, \ldots, n_{k+1} - n_k - 1$ ($\le m_{k+1} - m_k - 1$). Next, we map $\bigcup_{n_k \le m < m_{k+1}} \{0,1\}^m$ onto $\{0,1\}^{n_{k+1}-1}$. This way, F is onto and length-equality preserving. In more detail yet, when $j > i$ we map $\{0,1\}^j$ onto $\{0,1\}^i$ by $(x_1, \ldots, x_i, x_{i+1}, \ldots, x_j) \mapsto (x_1, \ldots, x_i)$. This way, $F : D_k \to R_k$ consists of projections.

Let us check that overall, F is polynomially balanced (in fact, input sizes and output sizes bound each other linearly): Indeed, F maps length m_k to length n_k, with $m_k < 6n_k$. Also, length $m_k + i$ is mapped to $n_k + i$ for $1 \le i < n_{k+1} - n_k$, with $m_k + i < 6n_k + i$. Finally, lengths between $m_k + n_{k+1} - n_k$ and $m_{k+1} - 1$ are mapped to length $n_{k+1} - 1$, with $m_{k+1} - 1 < 6n_{k+1} - 1$.

We see that F can be computed by a linear-size non-uniform family of circuits: For inputs of length m_k (for some k) we use the circuits C_k; for the other inputs, F is a projection.

Finally, let us check that no inverse F' of F is computable by a polynomial-size circuit family (if $\Pi_2^{\mathsf{P}} \neq \Sigma_2^{\mathsf{P}}$). The set $\{C_k : k \in \mathbb{N}\}$ that we constructed is infinite and $C_k \in \mathbf{C}_k$; hence any family $(C'_k : k \in \mathbb{N})$ of circuits that computes an inverse F' will satisfy $|C'_k| > |C_k|^k + k$ for all k. Since the set $\{n_k : k \in \mathbb{N}\}$ is infinite, the restriction of F to $\bigcup_{k\in\mathbb{N}}\{0,1\}^{m_k} \to \bigcup_{k\in\mathbb{N}}\{0,1\}^{n_k}$ has no inverse with size bounded by a polynomial (of fixed degree). Thus F has no polynomial-size inverse. □

4.2 A uniform one-way function

A result of Boppana and Lagarias [2] (combined with the Karp-Lipton theorem [9]) states that if $\Pi_2^{\mathsf{P}} \neq \Sigma_2^{\mathsf{P}}$ then there exists a function f that is one-way in the sense that f computable by a polynomial-size family of circuits, but the inverses of f are not computable by any polynomial-size family of circuits. The one-way functions considered in [2] are not polynomially balanced; moreover, they are either not total or not length-equality preserving (in the terminology of [2], the output can be the single symbol #). Also, these one-way functions are based on the Karp-Lipton theorem, so they are (apparently) not computable in uniform polynomial time. We will now construct a length-preserving function f that can be computed uniformly in polynomial time, but no inverse f' has a polynomial-size family of circuits.

We can describe any circuit C by a bitstring $\mathsf{code}(C)$, i.e., there is a "Gödel numbering" for circuits. Naturally there is also a decoding function $\mathsf{decode}(.)$ which is an inverse of $\mathsf{code}(.)$, i.e., $\mathsf{decode}(\mathsf{code}(C)) = C$. We can extend $\mathsf{decode}(.)$ to a total function, so any bitstring is decoded to a circuit. The encoding function $\mathsf{code}(.)$ is associated with an *evaluation function* ev such that

$$\mathsf{ev}(\mathsf{code}(C),\ x)\ =\ C(x) \quad \text{for all } x \in \{0,1\}^{m_C}.$$

Here we denote the length of the inputs of C by m_C and the length of the outputs by n_C. The functions $\mathsf{code}(.)$, $\mathsf{decode}(.)$, and $\mathsf{ev}(.,.)$ can be constructed so that they have special properties. The existence and the main properties of $\mathsf{ev}(.,.)$ and $\mathsf{code}(.)$ are well-known folklore, but we prove them here nevertheless because we will need detailed size and complexity estimates (items 3, 4, and 5 in the Proposition below).

Proposition 4.5 *Let* $\mathbf{C}$ *denote the set of all circuits. There exist functions*

$$\mathsf{code} :\ \mathbf{C} \to \{0,1\}^*,$$
$$\mathsf{decode} :\ \{0,1\}^* \to \mathbf{C},$$
$$\mathsf{ev} :\ \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*, \quad \textit{such that}$$

(1) *for all* $C \in \mathbf{C}$: $\mathsf{decode}(\mathsf{code}(C)) = C$;

(2) *for all* $c, x \in \{0,1\}^*$ *with* $|x| = m_{\mathsf{decode}(c)}$: $\mathsf{ev}(c,x) = [\mathsf{decode}(c)](x)$;
in particular, for all $C \in \mathbf{C}$, $x \in \{0,1\}^{m_C}$: $\mathsf{ev}(\mathsf{code}(C),\ x) = C(x)$;

(3) *for all* $C \in \mathbf{C}$: $|C|\log_2|C| < |\mathsf{code}(C)| < 6|C|\log_2|C|$;

(4) $\mathsf{decode}(.)$ *and* $\mathsf{ev}(.,.)$ *are total functions;*

(5.1) *the language* $\mathsf{im}(\mathsf{code}) = \mathsf{im}(\mathsf{code} \circ \mathsf{decode}) \subseteq \{0,1\}^*$ *belongs to* P;

(5.2) $\mathsf{code} \circ \mathsf{decode}(.) : \{0,1\}^* \to \{0,1\}^*$ *is polynomial-time computable and polynomially balanced;*

(5.3) $\mathsf{ev}(.,.)$ *is polynomial-time computable.*

Proof. We denote the sets of vertices and edges of C by V, respectively E. To construct the bitstring $\mathsf{code}(C)$ from a circuit C we first use a four-letter alphabet $\{a,b,c,d\}$. We label the vertices of the acyclic digraph of C injectively by strings over $\{a,b\}$, using binary numbering (with $a = 0$, $b = 1$), according to the order of V, from number 0 through $|V| - 1$. Each vertex is thus represented by a string in $\{a,b\}^*$ of length $\lceil \log_2 |V| \rceil$. In addition, each vertex is labeled by its gate type (namely and, or, not, fork, $\mathsf{in}_1, \ldots, \mathsf{in}_m$, $\mathsf{out}_1, \ldots, \mathsf{out}_n$), according to the gate map; strings over $\{c,d\}$ of length $\lceil \log_2(4+m+n) \rceil$ are used for these gate labels. As we alternate between

$\{a,b\}$ and $\{c,d\}$, no separator is needed. Thus we have a description of length $|V|(\lceil\log_2|V|\rceil + \lceil\log_2(4+m+n)\rceil)$ for the list of vertices and their gate types. Each edge is described by a pair of vertex codes, separated by a letter c, and any two edges are separated by a letter d. Thus the list of edges is described by a string of length $|E|(2+2\lceil\log_2|V|\rceil)$. So, $|\mathsf{code}(C)| = |V|(\lceil\log_2|V|\rceil + \lceil\log_2(4+m+n)\rceil) + |E|(2+2\lceil\log_2|V|\rceil)$. Hence $|\mathsf{code}(C)| > \frac{1}{2}|C|\log_2|C|$ (since $|V|^2 + |V| \geq |E| + |V| = |C|$), and $|\mathsf{code}(C)| < 3|C|\log_2|C|$ (since $|C| = |V| + |E|$). Turning $\mathsf{code}(C)$ into a bitstring (e.g., by encoding a, b, c, d as 00, 01, 10, 11, respectively) doubles the length. This completes the definition of $\mathsf{code}(.)$ and proves property (3).

To define the function decode we first let $\mathsf{decode}(\mathsf{code}(C)) = C$. When c is not the code of any circuit, we let $\mathsf{decode}(c)$ be the largest identity circuit (i.e., computing the identity map on $\{0,1\}^m$, for some m) with a code of length $\leq |c|$. This makes $\mathsf{decode}(.)$ a total function; property (1) also follows immediately.

An evaluation function ev can now be defined, based on the above construction of $\mathsf{code}(.)$ and $\mathsf{decode}(.)$. For any $(c,x) \in \{0,1\}^* \times \{0,1\}^*$, let $C = \mathsf{decode}(c)$. If $|x| = m_C$ then we define $\mathsf{ev}(c,x) = [\mathsf{decode}(c)](x)$. If $|x| \neq m_C$ we define $\mathsf{ev}(c,x) = x$. Properties (2) and (4) now hold.

The definitions of code and decode make it easy to check whether a string c is an encoding of a circuit, and to decode c (or to generate an identity circuit if c is not a code). The inequalities in (3) imply that $\mathsf{decode} \circ \mathsf{code}(.)$ is polynomially balanced. This shows properties (5.1) and (5.2). The definitions of code, decode, and ev make it easy to compute $\mathsf{ev}(c,x)$, so we have (5.3). The details are very similar to the proof that the circuit value problem is in P (see section 4.3 of [14]). □

The function ev is neither length-equality preserving nor polynomially balanced.

Proposition 4.6 *Let m and n denote, respectively, the number of input and output vertices of a circuit C. Theorem 3.4 still holds when one only considers circuits C that satisfy $|C| < 2m$ and $m = n$ (i.e., the function $C(.)$ is length-preserving).*

Theorem 3.4 also holds when one only considers circuits C with $m = n$ and $|\mathsf{code}(C)| < 12\,m\log_2(2\,m)$.

Proof. From C one can construct a circuit C_1 with equal numbers of input and output vertices. If $m < n$ one adds $n - m$ extra input vertices that are not connected to anything else in the circuit. If $m > n$ one adds $m - n$ new output vertices that carry the constant boolean value 0. A constant 0 can be created by making two copies of the input x_1 (by forking twice) and then taking $x_1 \wedge \overline{x_1}$ (= 0); this uses 4 gates and 6 wires. Making $m - n - 1$ more copies of 0 uses $m - n - 1$ fork gates and $2(m - n - 1)$ more wires. Now $m_1 = n_1 = \mathsf{max}\{n, m\}$, and $|C_1| \leq |C| + 3|m - n| + 10$ (where $|m - n|$ denotes the absolute value of $m - n$). Inverting C is equivalent to inverting C_1.

In any circuit C_1 one can add $|C_1|$ identity wires. An identity wire has two vertices and one edge, so the resulting circuit C_2 has size $|C_2| = 4|C_1|$, and $m_2 = m_1 + 3|C_1|$ input vertices, and $n_2 = n_1 + 3|C_1|$ output vertices. Hence, $|C_2| < m_2 + n_2$. Recall that circuit size is defined to be the number of vertices plus the number of edges in the circuit. If $m_1 = n_1$ then $m_2 = n_2$, and $|C_2| < 2\,m_2$. Since C_1 and C_2 differ only by identity wires, there is a one-to-one correspondence between inverses of C_1 and of C_2; an inverse for C_2 can be obtained from an inverse of C_1 by adding identity wires; an inverse for C_1 can be obtained from an inverse of C_2 by removing the extra identity wires.

By Prop. 4.5, C_2 also satisfies $|\mathsf{code}(C_2)| \leq 6|C_2|\log_2|C_2|$. We saw that $|C_2| < 2\,m_2$, hence $|\mathsf{code}(C_2)| < 12\,m_2\log_2(2\,m_2)$.

The circuits C_1 and C_2 can be constructed from C deterministically in polynomial time. Moreover, an inverse of C can be obtained in polynomial time from an inverse of C_1 or C_2, and vice versa. Hence, C has an inverse of size $\leq p(|C|)$ (for some polynomial $p(.)$) iff C_i has an inverse of size $\leq p_i(|C|)$ (for some polynomial $p_i(.)$), $i = 1, 2$. Since the existence of polynomial-size inverses for all circuits C implies $\Pi_2^{\mathsf{P}} = \Sigma_2^{\mathsf{P}}$ (by Theorem 3.4), the existence of polynomial-size inverses for circuits C_i also implies $\Pi_2^{\mathsf{P}} = \Sigma_2^{\mathsf{P}}$. □

Based on Propositions 4.5 and 4.6 we now construct a function which is one-way by circuit size. We start with the function

$$\mathsf{ev}_{\mathsf{circ}} : \quad (c,\ x) \longmapsto (c,\ [\mathsf{decode}(c)](x))$$

which is just the pairing $\langle \pi_1, \mathsf{ev} \rangle$ of the first projection $\pi_1 : (x_1, x_2) \longmapsto x_1$ and the evaluation function ev. We saw that ev is a total function that can be computed deterministically in polynomial time, hence $\mathsf{ev}_{\mathsf{circ}}$ is also total and polynomial-time computable. Levin observed that $\mathsf{ev}_{\mathsf{circ}}$ is a complete or "universal" one-way function, for a certain definition of one-way functions and for certain reductions between functions (see [10], [11], [7], and [15]).

The function $\mathsf{ev}_{\mathsf{circ}}$ is *polynomially balanced*. Indeed, for any input $X = (\mathsf{code}(C), x)$ and output $Y = (\mathsf{code}(C), C(x))$ of $\mathsf{ev}_{\mathsf{circ}}$ we have: $|X| = |\mathsf{code}(C)| + |x| \ \leq\ 2\,(|\mathsf{code}(C)| + |C(x)|) = 2\,|Y|$, and $|Y| = |\mathsf{code}(C)| + |C(x)| \ \leq\ 2\,(|\mathsf{code}(C)| + |x|) = 2\,|X|$, using the facts that $|x| \leq |C|$, $|C(x)| \leq |C|$, and $|C| \leq |\mathsf{code}(C)|$. Also, if c is not the code of any circuit then $\mathsf{ev}_{\mathsf{circ}}(c, x) = (c, x)$, so length is preserved in that case.

The function $\mathsf{ev}_{\mathsf{circ}}$ is not length-equality preserving, therefore we introduce a special evaluation function $\mathsf{ev}_o : \{0,1\}^* \times \{0,1\}^* \longrightarrow \{0,1\}^* \times \{0,1\}^*$,

$$\mathsf{ev}_o(c, x) \ =\ \begin{cases} (c,\ C(x)) & \text{if } \ c = \mathsf{code}(C),\ \ |c| \leq 12\, m_C\, \log_2(2\, m_C), \text{ and } \ |x| = m_C = n_C\,, \\ (c, x) & \text{otherwise.} \end{cases}$$

This definition makes ev_o *length-preserving*, hence it is also length-equality preserving and polynomially balanced. Clearly, ev_o is also uniformly computable in polynomial time. The definition was made in such a way that Prop. 4.6 can be applied.

Lemma 4.7 *If* $\Pi_2^{\mathsf{P}} \neq \Sigma_2^{\mathsf{P}}$ *then the special evaluation function* ev_o *is one-way by circuit size.*

Proof. By contraposition, let us assume that ev_o has an inverse function ev'_o which is computed by a polynomial-size family of circuits $\mathbf{E}' = (E'_i : i \in \mathbb{N})$. So, there is a polynomial $p(.)$ such that for all i, $|E'_i| \leq p(i)$. The circuit E'_i takes inputs of the form $(c, y) \in \{0,1\}^* \times \{0,1\}^*$ with $i = |c| + |y|$. Consider the case where $c = \mathsf{code}(C)$ for any circuit C such that $m_C = n_C = |y|$, and $|c| \leq 12\, m_C\, \log_2(2m_C)$. Then $i = |c| + n_C = |c| + m_C$. We let $C' = E'_i(\mathsf{code}(C),\ \cdot)$; this is the circuit E'_i with the c-input hardwired to the value $\mathsf{code}(C)$. Then the existence of an inverse C' for every circuit C as in Prop. 4.6, implies $\Pi_2^{\mathsf{P}} = \Sigma_2^{\mathsf{P}}$. □

Lemma 4.7 immediately implies:

Theorem 4.8 *If* $\Pi_2^{\mathsf{P}} \neq \Sigma_2^{\mathsf{P}}$ *then there exist length-preserving functions that are* one-way by circuit size *and computable uniformly in polynomial time.* □

References

[1] J.C. Birget, "The $\mathcal{R}$- and $\mathcal{L}$-orders of the Thompson-Higman monoid $M_{k,1}$ and their complexity", *International J. of Algebra and Computation*, 20.4 (June 2010) 489-524.

[2] R. Boppana, J. Lagarias, "One-way functions and circuit complexity", *Information and Computation*, 74.3 (1987) 26-240.

[3] A. Borodin, A. Demers, "Some comments on functional self-reducibility and the NP hierarchy", Technical Report TR 76-284, Dept. of Computer Science, Cornell University (July 1976).

[4] H. Buhrman, L. Fortnow, M. Koucký, J. Rogers, N. Vereshchagin, "Inverting onto functions and the polynomial hierarchy", *Theory of Computing Systems*, 46.1 (2010) 143-156.

[5] D.Z. Du, K.I. Ko, *Theory of computational complexity*, Wiley (2000).

[6] S. Fenner, L. Fortnow, A. Naik, J. Rogers, "Inverting onto functions", *Information and Computation*, 186 (2003) 90-103.

[7] O. Goldreich, *Foundations of Cryptography, Basic Tools*, Cambridge U. Press (2001).

[8] L. Hemaspaandra, M. Ogihara, *The complexity theory companion*, Springer (2002).

[9] R.M. Karp, R.J. Lipton, "Some connections between nonuniform and uniform complexity classes," *Proc. 12th ACM Symposium on Theory of Computation (STOC)*, (1980) 302-309. Journal version: "Turing machines that take advice", *L'Enseignement Mathématique*, 28 (1982) 191-201.

[10] L. Levin, "One-way functions and pseudo-random generators", *Combinatorica* 7.4 (1987) 357-363.

[11] L. Levin, "The tale of one-way functions", *Problemy Peredatshi Informatsii*, 39(1):92-103, 2003.
http://arxiv.org/abs/cs.CR/0012023

[12] E.H. Moore, "On the reciprocal of the general algebraic matrix", *Bulletin of the American Mathematical Society*, 26 (1920) 394-395.

[13] J. von Neumann, "On regular rings", *Proc. of the National Academy of Sciences of the USA*, 22 (1936) 707-713.

[14] Ch. Papadimitriou, *Computational Complexity*, Addison-Wesley (1994).

[15] L. Trevisan, "The program-enumeration bottleneck in average-case complexity", TR10-034 (March 2010) http://www.eccc.uni-trier.de/report/2010/034

[16] I. Wegener, *The complexity of boolean functions*, Wiley/Teubner (1987).

Jean-Camille Birget
Dept. of Computer Science
Rutgers University at Camden
Camden, NJ 08102, USA
birget@camden.rutgers.edu